Autonomous Agents and Multi-Agent Systems (2021) 35:16
https://doi.org/10.1007/s10458-021-09499-6

# AVDM: A hierarchical command-and-control system architecture for cooperative autonomous vehicles in highways scenario using microscopic simulations

Thomas Braud[1] · Jordan Ivanchev[1,2] · Corvin Deboeser[1] · Alois Knoll[2,3] · David Eckhoff[1,2] · Alberto Sangiovanni-Vincentelli[4,5]

Accepted: 22 March 2021 / Published online: 3 April 2021


## Abstract

Microscopic agent-based traffic simulation is an important tool for the efficient and safe resolution of various traffic challenges accompanying the introduction of autonomous vehicles on the roads. Both the variety of questions that can be asked and the quality of answers provided by simulations, however, depend on the underlying models. In mixed traffic, the two most critical models are the models describing the driving behaviour of humans and AVs, respectively. This paper presents AVDM (Autonomous Vehicle Driving Model), a hierarchical AV behaviour model that allows the holistic evaluation of autonomous and mixed traffic by unifying a wide spectrum of AV functionality, including long-term planning, path planning, complex platooning manoeuvres, and low-level longitudinal and lateral control. The model consists of hierarchically layered modules bidirectionally connected by messages and commands. On top, a high-level planning module makes decisions whether to join/form platoons and how to follow the vehicle's route. A platooning manoeuvres layer guides involved AVs through the manoeuvres chosen to be executed, assisted by the trajectory planning layer, which, after finding viable paths through complex traffic conditions, sends simple commands to the low-level control layer to execute those paths. The model has been implemented in the BEHAVE mixed traffic simulation tool and achieved a 92% success rate for platoon joining manoeuvres in mixed traffic conditions. As a proof of concept, we conducted a mixed traffic simulation study showing that enabling platooning on a highway scenario shifts the velocity-density curve upwards despite the additional lane changing and manoeuvring it induces.

**Keywords** Autonomous vehicle · Vehicle platooning · Car following model · Microscopic traffic simulation

✉ Thomas Braud
thomas.braud@tum-create.edu.sg

1 TUMCREATE, Singapore, Singapore

2 Technische Universität München, Munich, Germany

3 Nanyang Technological University, Singapore, Singapore

4 BEARS, Singapore, Singapore

5 University of California, Berkeley, CA, USA

## 1 Introduction

Throughout the last three decades we have witnessed an exponential increase in efforts for developing fully automated vehicle technologies. Reliability and safety of autonomous vehicles (AVs) system are continuously improving with the final objective to integrate them into transportation systems. This integration, however, presents multiple challenges that need to be carefully analysed. One of the main challenges is often referred to as mixed traffic: since conventional vehicles will not be replaced by fully automated vehicles overnight, humans and autonomous vehicles will have to share the roads for a period of time. The implications of this new traffic environment are not trivial to predict.

Real life deployments and tests have resulted in numerous accidents, unfortunately also including fatalities [62]. Governments and regulatory bodies have become increasingly concerned over the safety of autonomous vehicles. A reliable confidence level for the safety properties of AVs can be reached only after testing specific critical scenarios and driving hundreds of millions, or even billions of kilometres [52]. The large amounts of time and resources required, combined with the risk exposure to have more accidents calls for alternatives. A feasible option would be to move some of these tests from the real-world to a simulation environment, decreasing drastically the required time, resources, and casualty risks.

There are multiple simulation options for the evaluation of mixed traffic scenarios. Different abstraction levels and testing setups (also referred to as "X"-in-The-Loop) can be considered. Each level of abstraction refers to the degree of details each component of the AV system is modeled after. For example, simulation tools such as VIPS [34], CARLA [21], Autoware.ai [54] and ROS [1, 72] include a physical engine as well as models for sensors and mechanical components.

However, analysis with a cyber-physical platform is complex and usually favored for the latest stage of the development process. Moreover, simulating sensors, actuators, and mechanical components, as well as the implementation of the complete software, requires a considerable amount of computation. It is therefore, highly challenging to increase the amount of vehicles in the simulation to this level of detail. Simulating traffic on a larger scale requires a higher level of abstraction.

Macroscopic models provide a system-wide picture of traffic based on a number of assumptions and simplifications. Their low computational requirements make them attractive for city-scale studies. In fact, studies have been carried out utilizing macroscopic simulation to evaluate the implications of mixed traffic [9, 50]. However, behavioural aspects such as lane changing manoeuvres, and, in fact, all types of interactions between vehicles are not accounted for in these macroscopic models. Analysing the safety of AVs in traffic is therefore impossible due to the lack of vehicle interactions being simulated. This issue can be tackled by using microscopic agent-based simulation instead [41].

In microscopic simulation, a vehicle is represented by an agent governed by behaviour models that interact with the environment. Multiple microscopic traffic simulators are available such as VISSIM [25], SUMO [63], or AIMSUN [5]. In this paper, we considered the mixed traffic analysis framework called Behaviour Evaluation of Human and Autonomous VEhicles (BEHAVE) as it provides suitable conditions for mixed traffic evaluation [49].

Another category of traffic simulation models, known as mesoscopic models, is recently gaining more popularity. These models describe the traffic at a high level of detail, however, the behaviour of the vehicles and their interactions are described at a lower level of detail. This aims to maintain the individual interactions of the microscopic models while preserving the computational efficiency of the macroscopic ones. Mesoscopic models can be found in existing multi-agent traffic simulators such as MATsim [42] or POLARIS [93].

An alternative platform known as GAMA [94] has also been considered for traffic simulation studies [4, 23]. It is based on an agent-oriented programming language (GAML) for modeling agents and environments.

In order to have both human-driven and AVs in a simulation, their respective behaviour models need to be designed. The current state-of-the-art in human driven car-following[30, 98] and lane-changing models[56] used in agent-based simulations does not sufficiently capture human errors caused by poor perception, distraction, and aggression. It even seems that the current state of the art for vehicle behaviour is more suited to model AVs to some extent. Therefore, new models for human behaviour must be developed. Treiber has extended the Intelligent Driving Model (IDM) [98] by including human reaction time and space-ahead anticipation in the Human Driver Model (HDM) [99] while Hamdar included human perception error in his model [35]. The lack of human factor research in the area of driver behavior has led Michon to propose a more cognitive approach [74]

Regarding AVs, the research community has already devoted plenty of resources to development various microscopic models like, which can be readily implemented in some simulation software such as PLEXE for SUMO [84], AIMSUN [2]. Commercial simulators such as PTV-VISSIM [33] also include models for AVs.

These microscopic models are, however, not sufficient to capture the entire spectrum of AV mobility since most of them do not include high-level planning and control needed for platoon formation [70, 76]. Plenty of resources have been consecrated to develop platoon models featuring join and split manoeuvres [13, 14, 29]. However, less attention was directed to develop platoon models for microscopic simulation. These platoon models do not perform any manoeuvres such as join or split [73] or they use simple methods where merge manoeuvres are performed when all participants are on the same lane [3]. In [20], a model has been proposed to describe how the flow rate and speed reduction (when vehicles are joining or leaving the platoon) affect the length of the platoon.

We believe it is significant that platoon formation is modeled since it has be shown platooning brings many benefits to traffic [55] such as (1) increasing road capacity and decreasing traffic congestion [40, 101], (2) reducing energy consumption [100], and (3) minimizing the chances of collision considering that the majority of road accidents are due to human errors [16].

It seems more likely that platooning will operate mostly on highways during the early stage of autonomous vehicle deployment. States like California or Florida have already authorized limited truck platooning testing [87]. Additional challenges come in when vehicles are platooning in more sophisticated environments such as intersections or traffic lights. For example, there is a risk that not all the platoon vehicles are able to cross the intersection when traffic light turn red too early. Recent research has investigated intersection management systems of autonomous vehicle [6, 103]. These systems improve travel delay at intersection while increasing throughput and reducing fuel consumption [60].

To enable the microscopic simulation of autonomous vehicles, we present a novel microscopic Autonomous Vehicle Driving Model (AVDM) based on a hierarchical command-and-control system. It enables the simulation and testing of different AVs functionality such as high-level decisions, cooperative behaviour (platooning), motion planning, and low-level control in a highway environment.

The main contributions of this paper are:

- AVDM, a hierarchical AV driving model for enabling platooning and long-term planning on a microscopic agent-based level.
- A microscopic agent-based study on effects of platooning in mixed traffic conditions on the velocity-density diagram.

The remainder of the paper is organized as follows: Sect. 2 presents literature related to simulating AVs and driving behaviour. Sect. 3 describes in detail the hierarchical command-and-control system used for our AV driving model. Finally, in Sect. 4, different experiments are performed to evaluate our autonomous vehicle driving model.

Table 1 provides the list of all parameters and variable used in this paper as well as their values if applicable. The values provided in this table are identical for every AVs.

## 2 Related work

Modeling more advanced human behavior for microscopic traffic simulation is, without a doubt, necessary. However, this is beyond the scope of the paper. Modeling autonomous vehicle driving behavior for microscopic simulation is the main focus.

Even with the rising interested in fully automated vehicles, the current literature on modeling autonomous vehicle driving behaviour for microscopic simulation is still somewhat limited. Existing studies in microscopic simulation mostly focus on modeling driving assistance systems such as Adapative Cruise Control (ACC) [70, 76]. As stated before, simulating fully automated vehicles requires to include other aspect such as perception and

**Table 1** List of all parameters and variable with their value if applicable

| Symbol | Parameter | Value |
|---|---|---|
| $v_i$ | Vehicle $i$ | |
| $x_i$ | Position of the vehicle $v_i$ | |
| $v_i$ | Velocity of the vehicle $v_i$ | |
| $a_i$ | Acceleration of the vehicle $v_i$ | |
| $U(v_i)$ | Utility function for the vehicle $v_i$ | |
| $a_j^i$ | Attribute $j$ for the vehicle $i$ | |
| $\eta_j$ | Normalization value for attribute $j$ | |
| $r$ | Maximum visible range | 100 m |
| $k_p$ | CC PI proportional gain | 1 |
| $k_i$ | CC PI int gain | 0.5 |
| $C_1$ | Weighting factor | 0.5 |
| $\omega_n$ | Controller bandwidth | 0.2 Hz |
| $\xi$ | Damping factor | 1 |
| $d_d$ | Desired gap for platoons | 5 m |
| $T_0$ | Time headway | 2 $s$ |
| $\lambda$ | ACC parameter | 0.1 |
| $l_i$ | Length of the vehicle $i$ | |
| $v_d$ | Desired velocity | |
| $v_\Delta$ | Maximum allowed velocity difference | 5 m.s$^{-1}$ |

high-level decision planning. Therefore, we believe that deriving a simulation model from the real architecture of an AV seems like a logical and viable option.

Such a general autonomous vehicle system representation is presented in a survey by Pendleton et al.[77] and shown in Fig. 1. This architecture is an abstraction from earlier developments [32] of autonomous vehicle systems [27, 111]. The system can be split into four modules represented as layers: Perception, Planning, Control, and Communication.

## 2.1 Perception

The *perception layer* refers to the ability of an AV to collect information and extract relevant knowledge from its surrounding using data from sensors and V2X messages. It can be split into two sub-layers defined as: Environmental perception and Localization.

The *Localization* or also referred as Simultaneous Localization and Mapping (SLAM) aims to build and update an unknown map while simultaneously tracking the AV's position and orientation within it. Multiple studies address this problem involving LIDARs [109, 110], cameras [15], cameras combined GPS and IMU [22, 102], or a combination of these sensors [66, 106].

An efficient *Environmental perception* requires to accurately detect and classify the different surrounding objects including : (1) moving and static obstacles (vehicle, pedestrians, bicycles, median...) position and velocity, (2) road geometry, (3) road signs information (traffic lights...). This can be achieved using Multi-Object Tracking (MOT) and segmentation methods that usually involve LIDARs [12, 86], cameras [53], radars [47] or a fusion between LIDARs and cameras [10, 83].

Implementing real sensor models in simulation (also called Hardware-In-the-Loop) has already been considered in several research works [21, 28, 34, 82]. However, using sensor models for larger-scale microscopic traffic simulation when the number of simulated vehicles increases may exceed the computation power requirement. Therefore, a perception model with a higher level of abstraction seems more suitable perception model for traffic simulation. In [105], Weyns introduced a generic perception model for situated multi-agent system where the agent's perception rely on a set of sensors, precepts (machine-readable real-object representation) and filters specific to the situation.

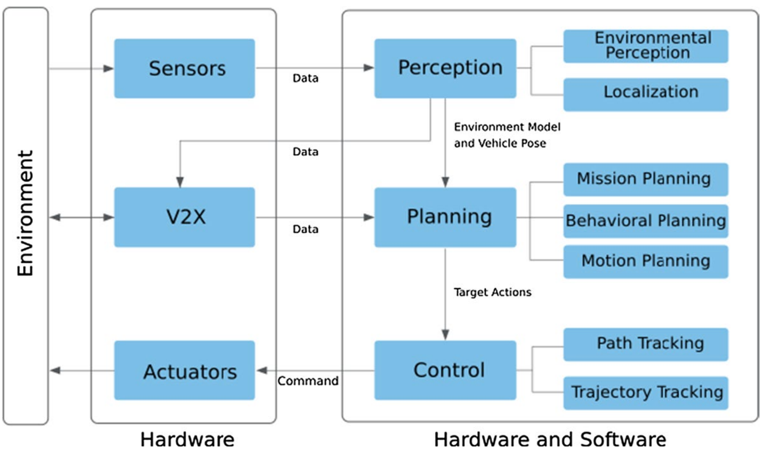



**Fig. 1** A typical autonomous vehicle system overview, highlighting core competencies (Based on: [77, 82])

This model inspired Ketenci to design a limited perception model for drivers in a traffic simulation [59]. Tabelpour and Mahmassani have proposed an alternative approach in [95] where sensor range and accuracy limitations were considered to create input data for the car-following model.

Considering the high number of agents, a more computationally efficient method is required to model the perception layer that can be included in microscopic simulation.

In this paper, we propose an alternative approach described in Sect. 3.

## 2.2 Planning

AVs are required to be able to handle any type of traffic condition by making safe and adequate driving decisions. This decision-making process is the core of the *planning layer*. Using the environment model provided by the *perception layer*, the *planning layer* aims to achieve specific goals such as (1) providing an efficient path plan that realises the mission incrementally, and (2) ensuring a safe journey that complies with all road regulations. The *planning layer* is presented as a compound of three sub-layers defined as: Mission planning, Behavioural planning, and Motion planning.

The *mission planner* considers high-level and long-term objectives, such as routing. It aims to provide the most efficient itinerary to reach a destination considering all pick-up/drop-off tasks. It is usually performed through graph search over a directed graph representing the road network, realised by using well-known algorithms such as Dijkstra [18] or A* [36]. For larger road networks, there exist a number of alternative methods with a focus on efficiency, as detailed in a survey paper on road planning [7].

The *behaviour planner* focuses on short-term objectives to ensure that a vehicle follows road regulations and safely interacts with other vehicles while progressing along the *mission planner*'s objectives. This allows AVs to make adequate driving decisions in city road traffic situations. Finite State Machine (FSMs) are usually utilised to dictate actions in response to a specific driving context [27].

In terms of platooning, research projects such as PATH [88] or SARTRE [81] have discussed a variety of platoon manoeuvres. FSMs are also considered for the definition and implementation of platooning manoeuvre [81]. However, these projects do not define manoeuvres and platoon entities (platoon leader, follower) formally. To provide a standardization of platoon manoeuvring, Maiti et al. have proposed a conceptual framework for platooning [71].

The *motion planner* layer's goal is to realise the *behaviour planner*'s decision. Motion planning refers to the process of generating a suitable path and deciding a set of actions to reach a specific target as represented in Fig. 2. Several approaches are based on using discrete representation of the environment such as Voronoi-diagrams [19] or a cell decomposition [26, 107]. Other approaches work with sampling-based methods such as Probabilistic RoadMaps [67] or Rapidly-exploring Random Trees [64]. More recent research apply the Artificial Potential Field (APF) method to the motion planning [43, 45]

The functionality of conventional microscopic driving models focuses on keeping a safe distance to other vehicles, maintaining a desired velocity, and making lane changing decisions. However, following a desired trajectory through traffic falls outside of their scope. Therefore, in order for AVs to execute commands of the *motion planning* layer, current driving models need to be extended.

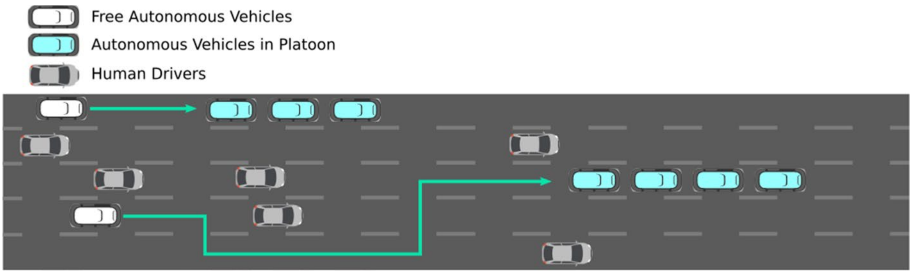



**Fig. 2** AVs receive information about a platoon and plan trajectories that includes multiple lane changes in order to safely join it

### 2.3 Control

The *control layer* guides the vehicle to execute the planned actions generated by the *planning layer* on a physical level. Control systems that autonomously drive the vehicle are usually divided into longitudinal and lateral control. In simulation, the longitudinal axis movement is described using car-following models which determine vehicle acceleration or deceleration depending on the velocity and position of vehicles in front. If there is no vehicle directly in front, for instance after changing lanes, longitudinal control allows the car to converge to a desired free-flow velocity. Numerous car-following model can be found in literature such as the Intelligent Driver Model [98] and its enhancement [58], Gipp's model [30] or the Optimal Velocity Model [37]. Other approaches can also be found based on Proportional Integral Derivative (PID) controllers [38, 68, 80] or Model Predictive Control (MPC)[61, 69].

Most of these car-following models can be used as reference for Adaptive Cruise Control (ACC) system's behavior since they are designed to maintain a safe gap to the vehicle ahead.

It has been shown that under specific parameter settings, ACC may lead to string instability [85]. Any variations caused by decelerating or accelerating vehicles will then be amplified, generating stop and go waves, or even accidents. For this reason, the research community started to work on an enhanced version named Cooperative Adaptive Cruise Control (CACC), which exploits wireless communication among vehicles. The idea is to share information such as acceleration, velocity, position, etc, to improve the responsiveness of the system. CACC also provides interesting and valuable features such as string stability for platooning [84]. Moreover, CACC allows a lower headway than ACC systems, effectively increasing road capacity. The literature reports several CACCs which differ in design, characteristics, and requirements [3, 45, 75, 79].

More recent research for collaborative driving consider communication delay [78, 97] in their approach, with the aim of keeping string stability despite the presence of unavoidable communication issues.

While the implications of integrating CACC in traffic systems have been widely studied, to the best of our knowledge, the impact of specific platooning (join/split) manoeuvres has not been studied in detail yet [40].

Lateral control determines if the vehicle should perform a lane change and is represented in simulation by lane-changing models. These models will consider information about adjacent vehicles and whether performing a lane change is beneficial for the changing vehicle. Unlike car-following models, only few lane-changing models have been proposed in the literature. The current state of the art for lane-changing model is MOBIL introduced in [57]. It takes as input the current velocity and acceleration of the vehicle and estimates the hypothetical velocities and accelerations the vehicle would have if it moved to the adjacent lanes, considering the potential preceding and following vehicles. Making this decision depends on the benefit for the vehicle and the weighted change of acceleration for the current follower and the new follower. This weight is called the politeness factor and represents the importance of not impacting other participants too much by changing lanes. MOBIL can be coupled with numerous car-following models such as IDM. Tabelpour and Mahmassani proposed another approach in [96] based on game theory that also exploits V2V communication.

In many simulations, lateral movement is performed instantaneously, which means the vehicle will not move continuously on the lateral axis but will be ’teleported’ to the target lane. Implementing and testing any cooperative manoeuvres is, therefore, constrained. Moreover, as stated in [40], impacts of lateral controllers on traffic throughput are usually not considered.

### 2.4 Communication

The *communication layer*, or Vehicle-to-X layer, allows AVs to exchange information. The information being exchanged is utilised by the *perception layer* (for additional information) and the *control layer* (for instructions). V2X communication can be realized through different wireless technologies. V2X communication is often realised using IEEE 802.11p WLAN [46]. Other technologies include C-V2X, that is, communication based on cellular networks, such as 4G [11] or 5G [8]. The evaluation of these communication systems is often carried out by means of simulation. Multiple network simulators are available, such NS-2 [48], NS-3 [39] and Veins [91] (see [90] for a survey of their capabilities). These network simulators can be coupled with mobility simulators for an increased degree of realism [51, 92, 104].

The system architecture as presented in Fig. 1 has been designed to perform the control of a single vehicle. In addition to a communication system between vehicles, a more advanced control structure is required to enable cooperative driving behaviour. As stated before, current state-of-the-art driving models (longitudinal and lateral control) are not sufficient for higher-level types of behaviour. Most related work focuses on one a specific layer (e.g., control or communication) without considering the other layers. We believe that in order to holistically simulate and study AVs in mixed traffic, a model is required that unifies all layers and their functionality.

## 3 Autonomous vehicle driving model

This paper presents a step towards unifying all layers of an autonomous vehicle system by introducing a framework based on a hierarchical command-and-control system as shown in Fig. 3. It combines the Planning, Control, and Communication layers into a single behaviour model, effectively enabling the simulation and analysis of complex platooning manoeuvres, long-term planning of microscopic AV agents, and efficient AV integration solutions for city-scale deployment. Similar to Fig. 1, our proposed system is composed of four different layers ranging from *High-Level decisions* to *Low-Level control*.

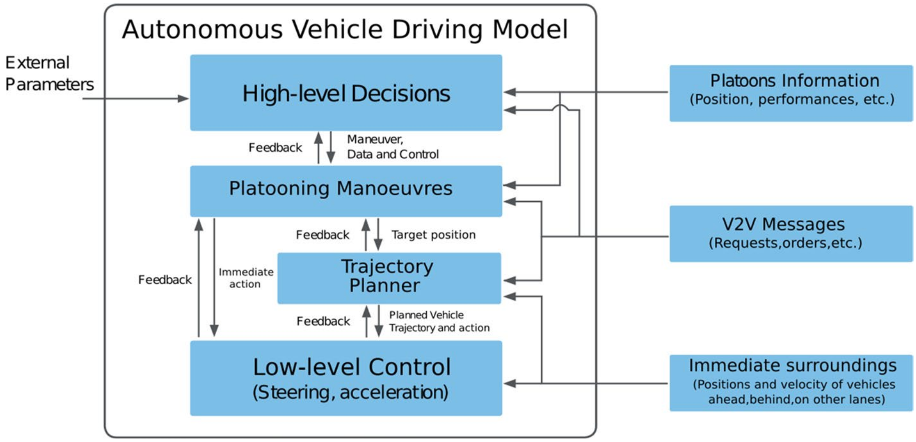



**Fig. 3** Hierarchical command-and-control structure for autonomous vehicles

As mentioned previously, platoons are more inclined to be deployed in highways before downtown. Focusing on highways as a first step seems a relevant option. It is also essential to indicate that we did not consider on-and-off ramps for this paper. The *High-Level decisions*,*Platooning Manoeuvres* and *Trajectory planner* are therefore built on this environment context. Every simulated AV in this paper is assumed to have the same vehicle characteristics (braking and acceleration) and share the same communication system and protocols.

The *Perception layer* as presented in Sect. 2.1 is not modeled in this paper. Consequently, no sensor models have been considered and implemented. However, the AVs still need to be aware of their surroundings to execute platoon manoeuvers safely. Therefore, we proposed an alternative approach to build the environment model required by our system. This alternative is assumed to be error-free as a first step. Modeling a perception system for microscopic simulation will be the focus of future work.

The topmost layer is the *High-Level decision* layer which can be interpreted as the behavioural layer from Fig. 1. It provides a decisions-making process regarding the choice of manoeuvres and actions to execute depending on the known environment. The *High-Level decision* layer is also responsible for reading and sending messages, evaluating requests, etc.

The *Platooning Manoeuvres* layer, a framework based on hierarchical state machines, guides vehicles to perform cooperative manoeuvres [17]. This framework ensures manoeuvres are executed correctly and, in cases where the manoeuvre has to be aborted, takes care of the safe termination of the procedure. The layer is connected upwards to the *High-Level decision* layer to allow the upper layer to also consider any incoming platooning messages layers and react accordingly. The downwards connection is used to provide different instructions in the form of action primitives to the *Trajectory planner* layer or directly to the *Low level control* layer.

The *Trajectory planner* generates trajectories, or paths, through current traffic to a new target position if required. (The *Trajectory planner* layer can be interpreted as the *motion planning* layer from Fig. 1.) The path will then be translated to a set of commands (e.g. acceleration, braking, lane change right/left) used as input to the *Low level control* layer, which is in charge of physically applying these commands.

Before describing all the layers that compose AVDM, we will first specify the vehicle dynamics model used in the microscopic traffic simulator BEHAVE [49].

### 3.1 Vehicle dynamics

Let $x_i, v_i, a_i$ be the longitudinal position, speed, and acceleration of the $i^{th}$ vehicle respectively, where $a_i$ is also the control input and $\tau > 0$ is the sampling time. The control $a_i$ is assumed to be constant on each time interval. The discrete-time longitudinal vehicle dynamics is described by the following double-integrator model:

$$\begin{cases} x_i(k+1) = x_i(k) + \tau v_i(k) + \frac{\tau^2}{2} a_i(k) \\ v_i(k+1) = v_i(k) + \tau a_i(k) \end{cases} \quad i = 1, ..., n \tag{1}$$

All the vehicle position and velocity are updated at each time step. This vehicle dynamics model is widely used for system-level car-following control design [31].

Since the model relies on messages being transferred between the vehicles, we will first introduce the nomenclature used in V2V communication.

### 3.2 Vehicle-to-Vehicle (V2V) communication

The syntax for V2V messages used by the AV communication system is inspired from Amoozadeh et al. [3]. Five different types of messages are defined (see Table 2): Requests (REQ), orders (ORD), done-confirmation (DN), abort (ABT), and acceptance/rejection (ACK/NACK). The types REQ, ORD, and DN are intended for specific manoeuvres. For example, a free AV seeking to join a platoon will send a REQ_JOIN message to the platoon leader.

In this paper we consider no packet loss or other communication restrictions, but assume a perfect communication channel under a unit-disc propagation model. Imperfections can also be modeled based on random process and added to our communication system.

### 3.3 Layer 1: High-level decisions

This layer allows vehicles to make decisions about future actions, for example, make the decision to form or join a platoon, initiate platoon lane-changes, etc. A vehicle can be either choosing a platoon to join/form (described in 3.3.1) and sending request messages or receiving a request message from another vehicle (described in 3.3.2). For this paper, it is assumed that request messages can only be sent by free vehicles, while vehicles that receive request messages can be either free (platoon formation) or platoon leaders (join manoeuvres).

#### 3.3.1 Evaluating platoons

A free vehicle $\tilde{v}$, willing to create or join a platoon evaluates its nearest neighbors to find the most suitable option. The utility of joining a considered vehicle or platoon $v_i$ is a combination of several attributes $a_j^i$ of the vehicle $i$: (1) difference in velocity, (2) path to vehicle, and (3) traffic density around the vehicle.

**Table 2** Message types definition. ORD, REQ, and DN are always combined with additional information. ABT are ACK/NACK are sent without any additions

| Sign | Type | Description | Example |
|---|---|---|---|
| REQ_[action] | Request | Message to request a specific action | A free vehicle sends a REQ_JOIN message to a platoon leader asking to join the platoon |
| ORD_[action] | Order | Message to order a vehicle to perform a (sub-)manoeuvre | A platoon is dissolved after its leader sends a ORD_DISSOLVE to all followers |
| DN_[action] | Done | Confirmation about the successful completion of a manoeuvre | A platoon follower sends a DN_GAPCLOSE after completing successfully |
| ABT | Abort | Information about the abort of a manoeuvre | A free vehicle not able to reach a position due to externals factors sends a ABT message to the platoon leader |
| ACK / NACK | Acceptance or Rejection | Accept (ACK) or refuse (NACK) a request | A platoon leader accepts the REQ_JOIN of a free vehicle by sending an ACK message |

The first attribute is, as indicated by its name, the absolute value of the difference in velocity between the considered vehicle $v_i$ and the free vehicle $\tilde{v}$. The *path to vehicle* refers to the difficulty of the autonomous vehicle $\tilde{v}$ to reach the target $v_i$. It is computed as the number of actions required to reach a target position (see Sect. 3.5) coupled with a malus system. *Traffic density* refers to the number of vehicles close to the considered vehicle $v_i$.

An interesting attribute to look at is the lifetime of a platoon, that is the expected time a vehicle spends inside the platoon (e.g. affected by the route choice of the vehicle). In this paper, traffic is simulated on an infinite highway without on/off ramps, therefore, the lifetime of a platoon is not a discriminating attribute.

The final utility of joining vehicle $i$ is represented by $U(v_i)$ which is the mean of the three attributes. This can be extended to an optimisation problem of finding the optimal weight for each attribute minimising the failure rate of the free vehicle reaching the considered vehicle $v_i$ in time.

$$U(v_i) = \frac{\sum_j f(a_j^i)}{3} \tag{2}$$

In order to be able to directly compare the attributes, they need to be normalised first. We achieve this by dividing the original attribute value by the maximum absolute value observed for this attribute $\eta_a$ in a set of experiments. After the normalisation we know that $-1 < a/\eta_a < 1$. Furthermore, in order to position this value between 0 and 1 we apply a Gaussian envelope to it via the function $f$ and further normalise the value as shown in Eq. 3 such that $0 < f(a) < 1$. A high value means the vehicle $v_i$ is a suitable candidate for platooning according to the attribute $a_j^i$ (e.g. $v_i$ is just in front or drive at the same speed):

$$f(a) = \frac{\exp\left(-\frac{1}{2}\left(\frac{a}{\eta}\right)^2\right) - \exp\left(-\frac{1}{2}\right)}{1 - \exp\left(-\frac{1}{2}\right)} \tag{3}$$

Figure 4 shows an example of how an AV evaluates its surrounding based on all the attributes described earlier. In this example, the blue vehicle is surrounded by three

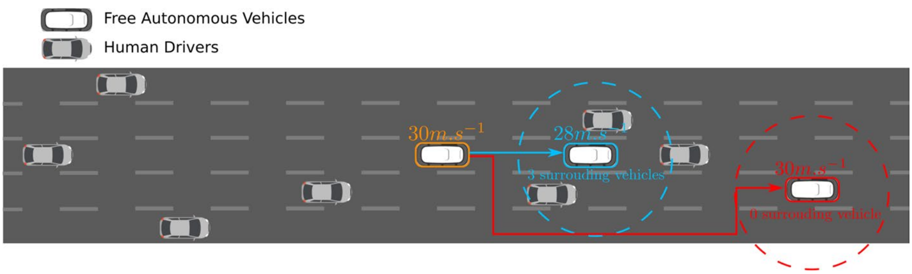



**Fig. 4** Illustration of how an AV in a free state (neither platooning nor performing manoeuvres) makes the most suitable choice for creating a platoon. The free vehicle $\tilde{v}$ (orange) is located at the center of the Figure. There are two other free autonomous vehicles (represented in blue and red). The remaining vehicles are human-driven. The orange AV compares the blue and red AV based on the three attributes (difference of velocity, path to vehicle and density). The blue AV seems closer and would therefore, be faster to reach. However, it is surrounded by several human drivers. Meanwhile the red AV is further away but less surrounded. The utility function $U(v_i)$ aims to determine the most suitable choice between the red and the blue AV

vehicles. We have observed through experimentation that the maximum absolute value $\eta_a$ for the density attribute is equal to 8. The Eq. 3 returns a value of:

$$f(a_{density}^{blue}) = \frac{\exp\left(-\frac{1}{2}\left(\frac{3}{8}\right)^2\right) - \exp\left(-\frac{1}{2}\right)}{1 - \exp\left(-\frac{1}{2}\right)} = 0.8274389$$

Applying this approach to the red vehicle will return 1 since there is no surrounding vehicles.

The vehicle $\tilde{v}$ will choose the option with the highest utility, given it is larger than a predefined threshold value $U_{th}$, otherwise no action will be taken. The influence of the threshold value is analyzed in Sect. 4.1. If a viable option is recognized, a REQ_JOIN message requesting to join/create a platoon is sent by vehicle $\tilde{v}$.

### 3.3.2 Evaluating a request message

Free vehicles and platoon leaders will accept any REQ_JOIN message. If a vehicle is already performing a manoeuvre or the platoon is at its maximum capacity, REQ_JOIN messages will be rejected. If the receiving vehicle is a free agent it will initiate a JOIN_TAIL; if it is a platoon leader, it has to decide what type of manoeuvre will be executed to join the platoon. REQ_JOIN messages are not sent to platoon follower vehicles. This decision-making process is described in Fig. 5. Accepting a request activates the *Platooning Manoeuvres* layer in charge of executing the chosen manoeuvre.

## 3.4 Layer 2: Platooning manoeuvres

The purpose of this layer is to safely perform the manoeuvre chosen by the *High-level Decisions* layer. To do so, this layer translates the manoeuvre into a set of comprehensible commands that will be executed by the *Low-Level control* layer. Any success or failure scenario is then reported to the *High-level Decisions*. Ideally, all autonomous vehicles share a catalogue of common manoeuvre descriptions to ensure interoperability of AVs from different providers.

Each manoeuvre consist of several sequential or parallel steps that need to be followed to arrive at a final, stable state. The most common method of describing manoeuvres is the use of Finite State Machines (FSM). For example, projects such as

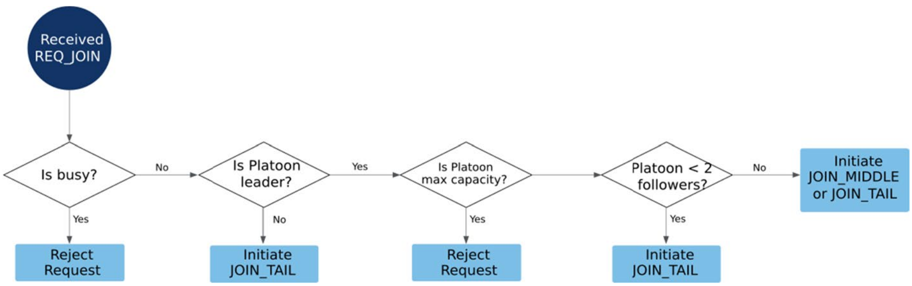



**Fig. 5** Decision making process when an AV that is not a follower in a platoon receives a REQ_JOIN

SARTRE [81] or SCANIA [65] use their own manoeuvre description based on FSMs. Without loss of generality, we have chosen to use the Standardisation, Encapsulation, Abstraction, and Decoupling (SEAD) framework as described in [17] to represent this layer. For the sake of comprehensiveness of the later sections in this article, we will give a brief introduction to this framework.

In SEAD, every manoeuvre is decomposed into sub-manoeuvres; sub- manoeuvres can be then represented as a sequence of logically ordered primitives. These primitives can be either be physical or logical. We refer to these actions as primitives as they represent the lowest level in the hierarchy and should be designed in a way so that they cannot be expressed by a combination of other primitives. Physical primitives provide directions that will affect the vehicle's position and velocity by communicating down to the *Low-Level control* layer. Logical primitives manage transition from one logical state (e.g. platoon leader, platoon follower) into another, or cause the vehicle to wait for an event to occur.

Since SEAD is based on FSM, every vehicle has to be in a certain state at any time. If the vehicle is currently not executing a platooning manoeuvre, it is in an Idle state. An Idle state does not refer to the physical state (location, velocity, acceleration, etc.), but a logical state that constitutes the driving mode it currently operates on (e.g. free vehicle, platoon leader). This logical state is used to determine the behaviour of a vehicle during a manoeuvre and its reaction to messages.

Sub-manoeuvres define the communication and actions between two or more vehicles while describing and encapsulating the behaviour for each participant. Sub-manoeuvres can perform transitions on the logical state of the vehicle. For instance, a free vehicle that has successfully joined a platoon will change its Idle state to platoon follower. These sub-manoeuvres are ultimately used in two kinds of overarching structures: The Reactive State Machine (RSM) and the Proactive Manoeuvring Engine (PME).

The RSM is in operation for any vehicle that participates in a manoeuvre in a reactive way, meaning that it will only react to incoming directives and report results. Thus, the RSM is a universal description of the reactive platooning-behaviour of vehicles and is thus identical for all vehicles. The PME, however, operates on the proactive participant. It contains the logic that defines the manoeuvres including any abort scenarios and provides directives to the reactive participants.

An example of this envisioned description language is given in Fig. 6 where we model the `JOIN_TAIL` manoeuvre. It contains three sub-manoeuvres, `MOVETOPOS`, `ATTACH`, and `GAPCLOSE`. The manoeuvre includes two vehicles, the platoon leader $v_{PL}$ and the free vehicle $v_{FV}$ willing to join the platoon.

First, the platoon leader (`PL`) sends the target position to the free vehicle. This triggers the `MOVETOPOS` sub-manoeuvre in which the joining vehicle tries to reach the given position by following a path among the traffic while `PL` waits for feedback.

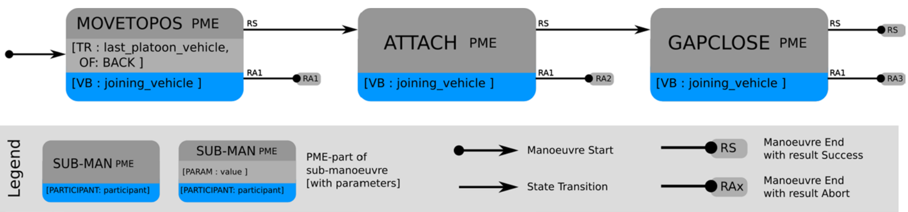



**Fig. 6** Description of the JOIN_TAIL manoeuvre according to the PME logic

MOVETOPOS can be aborted for two reasons: the free vehicle failed to reach the position (1) before a timeout or (2) because the target position is too far away.

In case MOVETOPOS succeeds, the next step is ATTACH where the joining vehicle changes its Idle state from Free Vehicle to Temporal Platoon Leader (TPL), forming a temporary platoon only containing itself.. Note, that this sub-manoeuvre is purely logical and does not trigger any physical action.

The last sub-manoeuvre of JOIN_TAIL is shown in detail in Fig. 7. The top part is the proactive manoeuvring engine (PME) of the platoon leader PL; the bottom part shows the reactive state machine (RSM) of the temporary platoon leader TPL. The smaller grey boxes represent the physical and logical primitives. PL sends a ORD_GAPCLOS message to TPL, ordering it to reduce its space ahead to a desired intra-platoon gap. After successfully reducing the gap, TPL changes its Idle state to Platoon Follower (PF). If closing the gap is taking too long, a timeout causes the platoon leader to abort the manoeuvre and sends an ABT message to the TPL.

In this relatively simple manoeuvre, all participants will already be in a stable state should any of the sub-manoeuvres be aborted. Manoeuvres with higher levels of complexity, however, may require an additional construction of sub-manoeuvres to handle each specific abort scenario. Defining a complete catalogue of platooning manoeuvres will be the focus of future work.

### 3.5 Layer 3: Trajectory planner

Some manoeuvres require that participating AVs move to specific positions (e.g. to the tail of a platoon) as fast as safely possible. The *Trajectory planner* layer aims to provide guidance on how to reach this specific position taking into consideration the current traffic conditions. This guidance is represented by a set of actions used as input to the *Low-level control* layer. The *Trajectory planner* layer can be thought of as the *motion planning* layer from Fig. 1. However, in order to plan its path, each vehicle needs to be constantly aware of

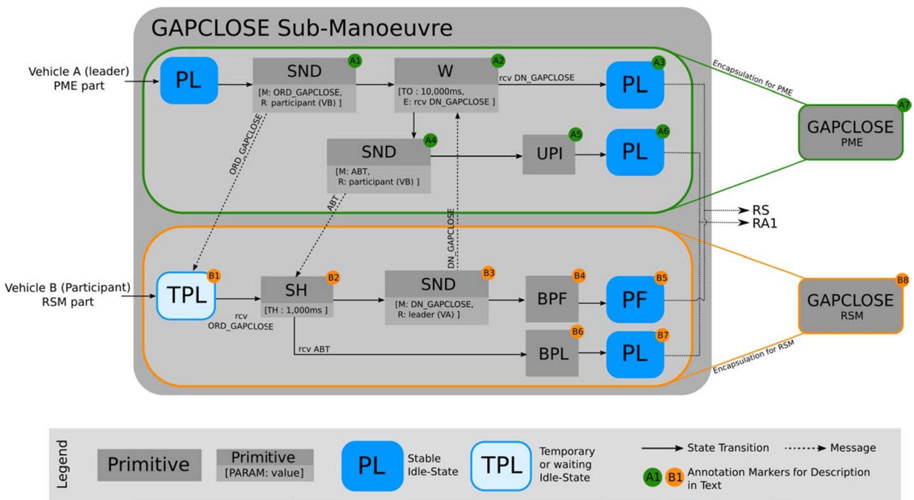



**Fig. 7** Example of sub-manoeuvre description and encapsulation for GAPCLOSE

its surrounding environment (positions and velocities of nearby vehicles) within a certain range, referred to as the neighbourhood range.

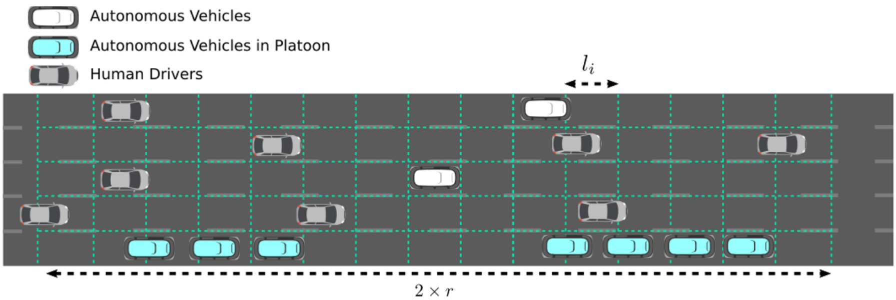
The neighbourhood range of every vehicle is extracted from an environment model that we have built using a modified version of the sweep and prune approach. The sweep and prune approach is well-known for its low computational cost for collision detection in physical simulations [24].

Instead of focusing on collisions, the sweep and prune is used here to locate the nearest neighbors within a certain range. Only one axis is sorted in this case since we are working in a infinite highway. The lateral position is therefore no longer a discriminating variable. Working on only one axis reduces the computation time compared to the general sweep and prune. However, it is not applicable to more complex road networks. Continuous $k$ nearest neighbors algorithms can be considered for larger road network [44].

The result is then used to create a grid network as represented in Fig. 8. Each cell is vehicle length long and one lane wide. Each cell also has a value which indicates what type of vehicle is contained in the cell (**H** for human-driven vehicle and **A** for autonomous vehicle). If two vehicles are located in one cell, the type of vehicle occupying more space within the cell will be considered.

At this point it must be noted that the environment model we are creating would typically first pass through a *perception layer* as described in Fig. 1. However, we consider this to be beyond the scope of this work, since any perception layer from the literature can simply be integrated into our proposed system as a function on the real input to the models provided by the simulation.

The grid shown in Fig. 8 is used as input to the path-finding A* algorithm [36]. Using information given by the platoon leader, A* provides the shortest path on the grid




**(a)** Neighbour search of an autonomous vehicle in the centre: The sweep and prune approach aims to locate all neighbors within a range define by the parameter $r$. Then all detected neighbours can be inserted into a grid.


**(b)** Positioning in the grid, "M" represents the position of the considered vehicle , "H" represents an normal vehicle driven by a human, "A" represents an autonomous vehicle.

**Fig. 8** Results of spatial partitioning for the green agent from the agent to his target. An example of a generated path is shown in Fig. 9. The path is then translated into a sequence of actions that can be: lane change left, lane change left, move forward which are sent to the *Low-level control* layer.

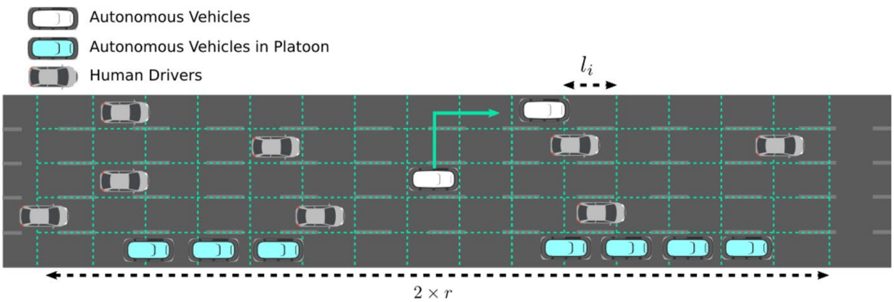
This sequence of actions is also used by the function $f(a)$ from Eq. 3 for the *path to vehicle* attribute. In the same example shown in Figs. 8 and 9, the result value is equal to 3.

Merging with a platoon is more challenging when several lane changes are involved. Moreover, the traffic flow might also be affected if the AV must slow down when its targeted vehicle is behind. Therefore, we have included a malus system. Every lane change will add a value of 2 to the attribute while decelerating will add a value of 1. The final value for the *path to vehicle* attribute in the example of Fig. 8b is equal to 7.

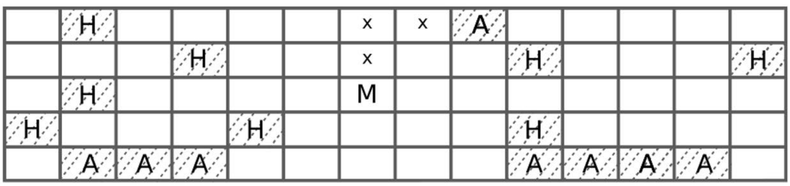
### 3.6 Layer 4: Low-level control

The *Low-level control* layer aims to apply any higher-level decisions on the physical level (acceleration or braking) required by a manoeuvre. This Section presents the modifications made to the current state of the art of longitudinal and lateral controllers to achieve this objective.

The low-level control layer is at the bottom of the hierarchy of the presented AV model. Its outputs are longitudinal and lateral accelerations that can be sent to any microscopic agent-based simulator engine to be applied during the next time step of execution.




**(a)** The A* algorithm will provide a path for the considered vehicle to follow. The path is displayed in green




**(b)** Grid information is used by A* to find the shortest way. An "x" represents the path the agent has to follow in order to reach the target

**Fig. 9** Results of path-finding for the VIP agent

### 3.6.1 Longitudinal controller

Usually referred to as car-following models, longitudinal controllers ensure that the vehicle keeps a safe distance to the preceding vehicle. As stated previously, conventional car-following models are not designed to apply the commands given by the *Trajectory Planner*. Therefore, we introduce a new longitudinal control logic to overcome this limitation. Our approach is a combination of three existing methods composed by CC, ACC, and CACC, all as defined in [80] where some modifications have been added to enable path following for AVs.

The control law used for the CC is defined as:

$$a_i = -k_p(v_i - v_d) - k_i \int (v_i - v_d)dt \tag{4}$$

where $v_i$ and $v_d$ are the current and the desired velocity, respectively. The parameters $k_p$ and $k_i$ can be changed to tune the behaviour of this controller.

The control law used for the ACC is defined as:

$$a_i = -\frac{1}{T_0}(v_i - v_{i-1} + \lambda(x_i - x_{i-1} + l_{i-1} + T_0 v_i)) \tag{5}$$

where $v_{i-1}$ is the velocity of the vehicle in front, $x_i$ and $x_{i-1}$ are the positions of the considered vehicle and its preceding vehicle.

The control law used for the CACC is defined:

$$a_i = \alpha_1 a_{i-1} + \alpha_2 a_0 + \alpha_3(x_i - x_{i-1} + l_{i-1} + d_d) + \alpha_4(v_i - v_0) + \alpha_5(v_i - v_{i-1}) \tag{6}$$

with,

$$\begin{aligned} \alpha_1 &= 1 - C_1;\ \alpha_2 = C_1;\ \alpha_5 = -\omega_n^2 \\ \alpha_3 &= -(2\xi - C_1(\xi + \sqrt{\xi^2 - 1}))\omega_n \\ \alpha_4 &= -C_1(\xi + \sqrt{\xi^2 - 1})\omega_n \end{aligned} \tag{7}$$

where $a_0$ and $v_0$ are the acceleration and velocity of the platoon leader; $d_d$ is the desired intra-platoon gap.

The value given to the desired gap $d_d$ is not fixed and may change during the simulation (e.g. during a `GAPCLOSE` sub-manoeuvre).

As illustrated in Fig. 10, the applied controller changes accordingly to the AV's current state.

To overcome the issue that these controllers are not capable of following a path or completing a task required by a manoeuvre, we extend them accordingly. "Forcing" the AVs to accelerate or decelerate is possible by controlling the desired velocity.

This new desired velocity considers the distance to the target vehicle. If the target vehicle is behind (or in front), the desired velocity will be slower (or faster) than the target vehicle's current velocity $v_t$. The desired velocity is defined as follows:

$$v_d(\varepsilon_t) = v_t + \frac{2v_\Delta}{\pi}\arctan(\alpha\epsilon_t) \tag{8}$$

where $v_\Delta$ represents the maximum allowed velocity difference to ensure smooth traffic and safety. It is defined as :

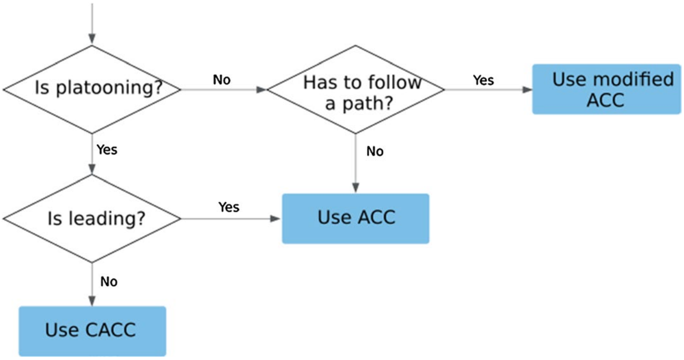



**Fig. 10** The autonomous vehicle current state defines which controller will be applied

$$v_\Delta = \frac{1}{10} v_t \tag{9}$$

The distance error function $\varepsilon_t$ is defined as:

$$\varepsilon_t = x_t - l_t - x_i - d_d \tag{10}$$

where $x_i$ and $x_t$ refer to the position of the considered AV and its target, $l_t$ refers to the length of the target vehicle, and $d_d$ refers to the desired gap.

Our *modified ACC* shown in Fig. 10 returns the minimum between (1) the output of Eq. (5) and (2) Eq. (4) with the new desired velocity from Eq. (8).

As shown in Fig. 12, the *modified ACC* converges to a desired position (as presented in Fig. 11) regardless whether the target vehicle is in front or behind.

### 3.6.2 Lateral controller

We have considered the current state-of-the-art lane-change model MOBIL [57] as a basis for our extensions. It can be modelled using the lane change utility function $u$ which has to satisfy a set of conditions:

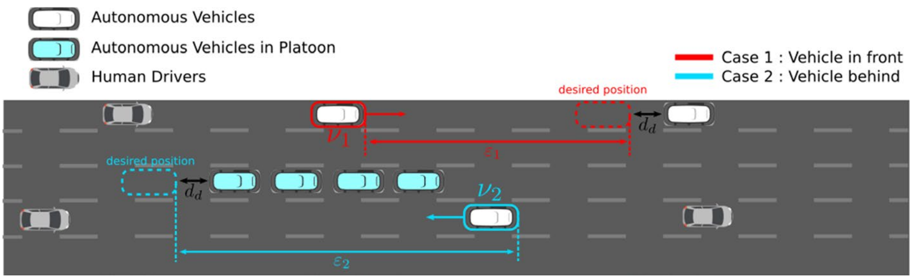
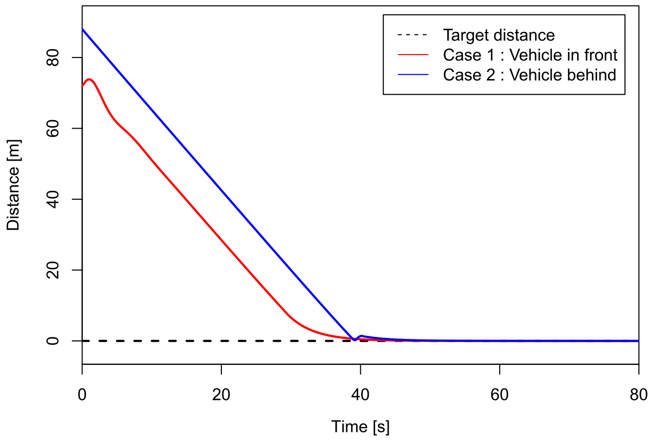
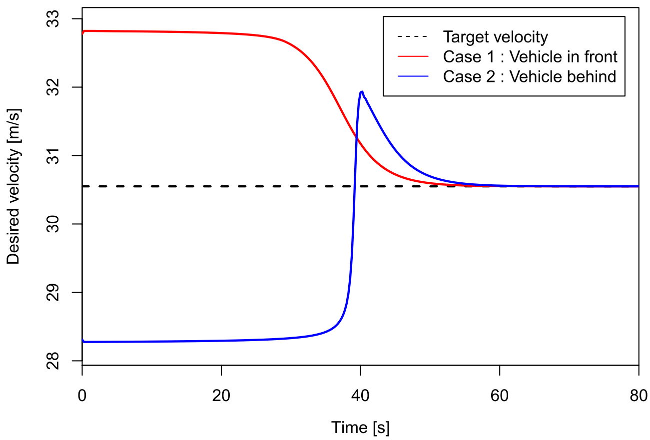



**Fig. 11** In this example, the *modified ACC* is evaluated in two scenarios. Case 1: target vehicle is the preceding vehicle of the considered AV ($v_1$), the objective is to reduce the space ahead towards a desired gap $d_d$ by accelerating. Case 2: target vehicle is located at the tail of a platoon behind the considered AV ($v_2$), the objective is to reach the desired position by decelerating




**(a)** Starting with an initial distance of 80m from the target, the vehicle manages to reach the desired position. (Only absolute values are plotted).




**(b)** Evolution of the desired velocity.

**Fig. 12** Two cases of a vehicle approaching a target vehicle using the modified ACC. In the first case, the joining vehicle is in front of the target vehicle, in the second case it is behind

$$u_L = \tilde{a}_i^L - a_i + P(\tilde{a}_n^L - a_n^L + \tilde{a}_o^L - a_o^L) - \Delta a_{th} - \Delta a_{bias} \tag{11}$$

$$u_R = \tilde{a}_i^R - a_i + P(\tilde{a}_n^R - a_n^R + \tilde{a}_o^R - a_o^R) - \Delta a_{th} - \Delta a_{bias} \tag{12}$$

where subscript $n$ represents a new following vehicle, $o$ is old following vehicle, and tilde represents the hypothetical value of acceleration when the vehicle has been placed on the respective adjacent lane. Whether a lane change is performed to the right or the left depends on the values of $C_R$ and $C_L$:

$$C_L = \begin{cases} 1, & If u_L > 0 \\ 0, & Else \end{cases} \tag{13}$$

$$C_R = \begin{cases} 1, & If u_R > 0 \\ 0, & Else \end{cases} \tag{14}$$

If both are 1, the values of $u_R$ and $u_L$ are compared. A vehicle chooses the lane with higher utility and initiates the according lane-change.

AVs participating in a manoeuvre will be commanded to perform a lane change that might not necessarily provide higher utility values according to the model. However, the lane change still needs to be executed for completing a manoeuvre or sub-manoeuvre. Therefore, a different lane change strategy is required. In order to ensure that a lane change will not have a disruptive effect on traffic, the AV will perform two tests which are similar to the functionality of gap-acceptance lane changing models.

First, the vehicle (platoon leader or free AV) evaluates the available space on the target lane. This available space is defined by the distance to the hypothetical preceding and following vehicle. If the space is higher than a threshold, performing a lane-change is considered safe. A second test is made by analysing the hypothetical deceleration of the potential new follower. The lane-change is considered safe and not disruptive to the flow of traffic, if the deceleration does not exceed a certain threshold, meaning the hypothetical new follower does not have to brake strongly. The lane-change will be performed if and only if the two conditions are satisfied. An example is shown in Fig. 13.

## 4 Evaluation

As a proof of concept for the functionality and usefulness of the AV model outlined in this paper, we have performed two experiments in mixed traffic conditions. The first experiment studies the success rate of the JOIN_TAIL manoeuvre while the second experiment investigates the effects AVs and platooning have on the velocity-density diagram of traffic flow.

We integrated our autonomous vehicle model into the mixed traffic analysis tool BEHAVE [49], based on the CityMoS engine [108]. It allows for the co-existence of

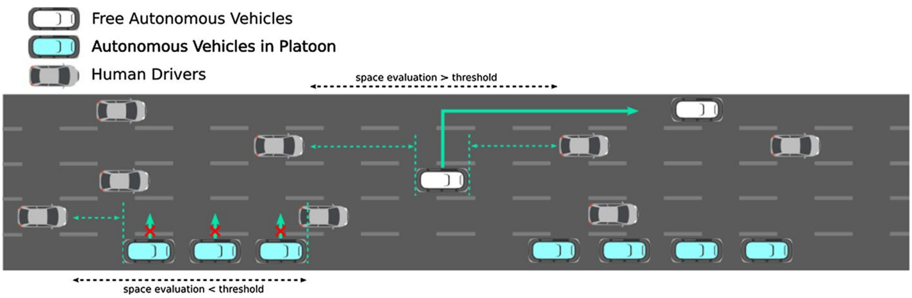



**Fig. 13** When an AV have to change lane to follow a defined path, it will make two tests. First it will check the space ahead and behind and make sure it does not go below a defined threshold. Second, it will estimate the hypothetical acceleration for the 'new' follower, if the deceleration is below another threshold, the lane change is considered dangerous and will not be performed

vehicles governed by different models on the road. Those groups of different vehicles are referred to as populations, and the model parameters of those populations can be further specified.

In both experiments, we have simulated a mixed traffic scenario of human-controlled vehicles and AVs. The Human Driver Model [99] is used to represent human behaviour and our AVDM governs the behaviour of AVs. The desired velocity $v_d$ for humans is drawn from a uniform distribution while all AVs share the same desired velocity, which is in the center of the uniform distribution used for humans. All other model parameters that the two populations have in common are identical. Table 3 shows the parameters value for the HDM used in this paper. The experiment conditions consist of a simulation region, centered around an ego vehicle, moving through an endless five lane highway. Since the highway does not contain on-and-off ramps, vehicles are continuously spawned/deleted at the borders of the simulated region. The spawning intensity can be adjusted in order to control the traffic density inside the simulated region. The traffic is assumed to be right-handed and all vehicles are allowed to overtake from both left and right sides. There is no dedicated AV lanes in this scenario.

### 4.1 JOIN_TAIL manoeuvre

This experiment studies the success rate of completing the `JOIN_TAIL` manoeuvre in mixed traffic. The traffic density for this experiment is kept constant at 0.02 vehicle/m/lane. The ratio between human vehicles and AVs is set to 1 : 1. A manoeuvre is considered successfully completed if and only if all the sub-manoeuvres that compose the manoeuvre are successful. For each simulation run, a total of 10, 000 manoeuvres have been performed.

The main AV model parameter which can be used to steer the dynamics of platooning in the simulation is the threshold utility value $U_{th}$ for triggering a platoon manoeuvre. Figure 14 shows the average success rate for completing `JOIN_TAIL` as a function of $U_{th}$. There seems to be only a marginal improvement in the success rate for utility values under 0.6. This could indicate that choosing to join a random platoon within the sensing area would result in 65% success rate for the `JOIN_TAIL` manoeuvre. Going beyond a certain critical utility value, however, (in this case 0.6) renders some of the options unacceptable and thus this is the area where the high level control layer effectively begins filtering out poor options.

**Table 3** Parameters of the HDM with the values used in this paper. (based on [99])

| Symbol | Parameter | Value |
|---|---|---|
| $T'$ | Reaction Time | 1 s |
| $n_a$ | Number of anticipated vehicles | 1 veh |
| $V_s$ | Relative distance error | 5% |
| $r_c$ | Inverse TTC error | 0.01/s |
| $\tau$ | Error correlation time | 20 s |
| $s_0$ | Minimum gap | 2 m |
| $T$ | Time head way | 2 s |
| $a$ | Maximum acceleration | 1.8 m.s$^{-2}$ |
| $b$ | Maximum deceleration | 2 m.s$^{-2}$ |

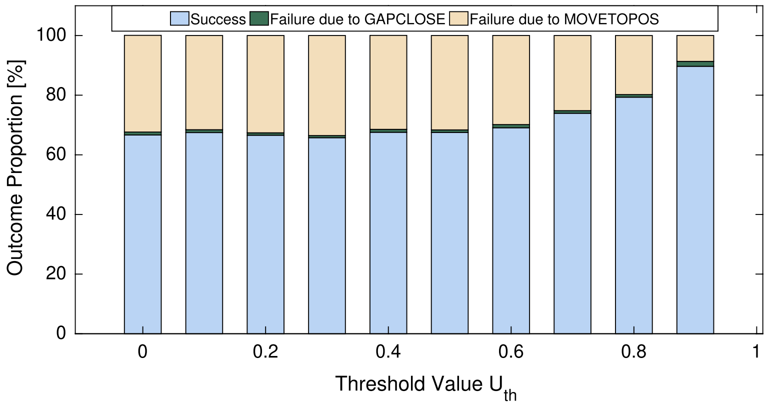



**Fig. 14** Influence of the threshold value $U_{th}$ on the outcome of the `JOIN_TAIL` manoeuvre

Failures are generally caused by 1) target that has been set but becomes hard to reach when the manoeuvre has already been initiated or 2) inability to execute a low-level command fast enough. The source of both failure types is fast changing traffic conditions. More than 80% of failures occur during the execution of the `MOVETOPOS` sub-manoeuvre and are due to external factors (e.g. slow vehicles preventing the sub-manoeuvre to be completed correctly).

Figure 15 shows the measured duration for every step of the `JOIN_TAIL` manoeuvre. With a success rate higher than 90%, the average time for an AV to complete the manoeuvre is less than 30*s*. The biggest variation is observed in the `MOVETOPOS` sub-manoeuvre due to its high sensitivity to traffic conditions surrounding the vehicle. The `ATTACH` sub-manoeuvre is purely logical and therefore has no time and no duration variation. The variation in the duration of the `GAPCLOSE` sub-manoeuvre is caused by the variation of the final position of the joining vehicle during the attach phase and also due to the varying traffic conditions in front of the platoon leader. The duration of this sub-manoeuvre can be

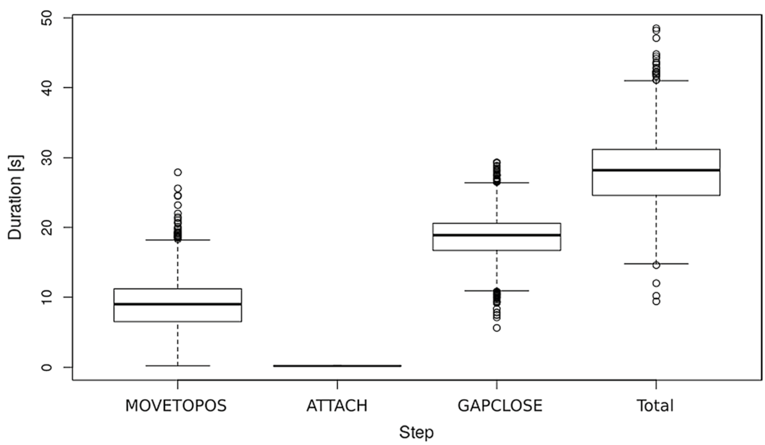



**Fig. 15** Duration of all sub-manoeuvres composing the `JOIN_TAIL` manoeuvre when they are successfully completed for $U_{th}$ set to 0.9

reduced tuning some of the car-following model parameters, however, a too quick GAP-CLOSE would induce a higher risk of collision if the platoon has to stop abruptly during the time the gap is being closed.

This experiment demonstrates two things : (1) the ability of the *Platooning manoeuvre* layer to supervise and execute successfully a manoeuvre or handling failures in mixed-traffic scenarios (2) the ability of the *Low-Level control* layer to follow a path and move to a specific position as mandated by the *Platooning manoeuvre* layer.

### 4.2 Average velocity with mixed traffic

In a second experiment, as a proof of concept, we study the velocity-density relationship of traffic for varying percentages of AVs on the road. For the purpose of this experiment, the spawning intensity in the simulation area is being gradually increased, thus producing traffic conditions from free flow to slight congestion. Furthermore, two settings of mixed traffic scenarios are defined: (1) platooning is disabled (2) platooning is enabled.

In the first setting, AVs are moving freely without initiating any cooperative manoeuvres, while in the second, AVs have the choice between creating/joining platoons or remaining free agents. The threshold utility value $U_{th}$ for triggering a platoon manoeuvre has been set to 0.8. For each experiment, we vary the proportion of humans and AVs in the simulation using 0, 25, 50, and 75 percent of AVs.

We can observe from the results shown in Fig. 16 that AVs introduce improvements to the average velocity of traffic in general. These improvements seem to be proportionate to the penetration rate of AVs in the simulation (vertical distance between the lines). Moreover, enabling platooning provides greater improvements to the average traffic flow (vertical distance between the dashed and solid lines of the same colour).

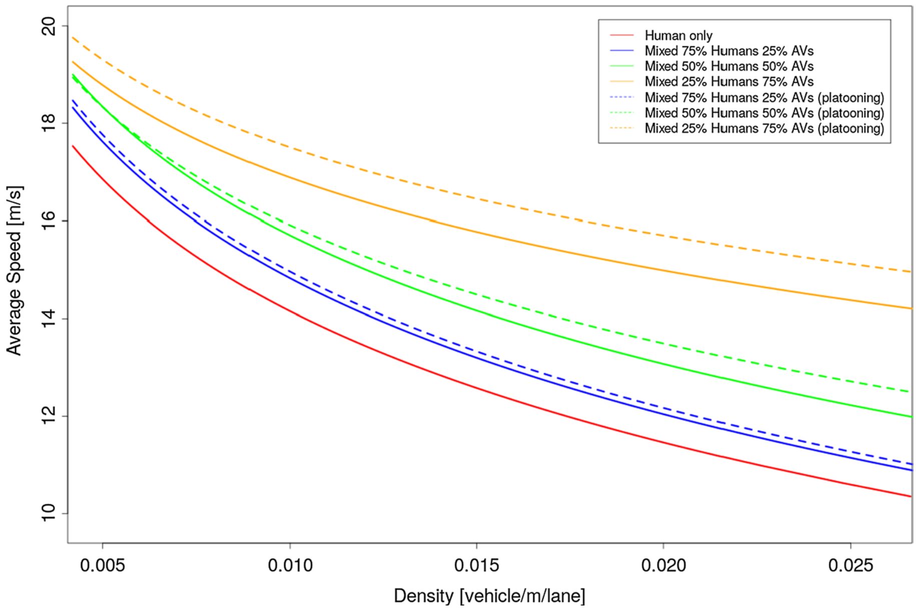



**Fig. 16** Velocity-density diagram comparison between different mixed traffic scenarios

These results are in-line with the general consensus that AVs can contribute to improving the average traffic flow [73, 89, 101]. This is mainly due to smaller inter-vehicle distances that technically increase the road capacity. However, we believe that more research is mandated to precisely quantify the benefits that can be expected from various platooning strategies. Many of the studies on the benefits of platooning found in the literature do not take into consideration possible disruptions caused by platooning manoeuvres. The present AVDM model can be used to study these effects in detail. In our proof-of-concept study, we can already observe that for simple join/tail manoeuvres, the positives outweigh the negatives in our simulation scenario. This is strong indication that AVDM is able to adequately guide the vehicles through the platooning process in mixed traffic conditions and adapt quickly to the fast changing traffic situations.

## 5 Conclusion

In this paper, we presented a hierarchically structured model for autonomous vehicles in microscopic agent-based simulations to enable more realistic mixed traffic simulation. The model incorporates a large spectrum of existing AV technology such as 1) long-term planning and decision-making, 2) complex platoon manoeuvre execution, 3) path planning, and 4) low level vehicle control. Furthermore, the model architecture takes care of the connections between those various control layers and determines the information flow in order to holistically simulate AVs.

As a proof of concept, we implemented the proposed model in the BEHAVE simulation platform and performed an experiment showing that platoon joining manoeuvres have a high success rate in mixed traffic. This indicates that our model successfully incorporates the respective processes on various levels of abstraction and effectively manages the flow of commands back and forth between the hierarchical layers. To demonstrate the usefulness of mixed traffic simulation, we also performed an experiment to test whether platooning would improve traffic conditions in a mixed traffic scenario. This experiment yielded positive results, showing that for the studied AV penetration rates, platooning always brings improvement to the traffic flow compared to scenarios with AVs that do not engage in cooperative behaviour.

In order to improve upon mixed traffic simulation both AV and human driver models need to be extended. A limitation of the experiments performed in this paper is the incorporated human driver model as it does not manage to contrast the human-driver behaviour from the AVs sufficiently. The main human factor that differentiates this model is a longer reaction time. However, longer reaction times, even with high levels of spatial anticipation, produce considerable stop-and-go waves and even accidents. In order to overcome this issue, future efforts include building on top of human driver models. We believe that human models should also emulate unpredictable and even irrational human behaviour. Including stochastic elements in the model is a possible approach to do so. This aims to bring them closer to reality in terms of mixed traffic simulation.

As future work we would like to extend the model by introducing statistical modeling of the perception layer and increasing the accuracy of V2X communication. The impacts of perception errors and sensor noises on the *Trajectory Planner* and *Low-level Control* will be investigated accordingly. The *High-Level decisions*, *Platooning Manoeuvres* and *Trajectory Planner* layers also need to be extended for more sophisticated environments such as intersections and traffic lights. Constrains can also be added in the *High-Level decisions* and *Trajectory Planner* to take different road regulations and policies into account. These will allow the model to represent a complete AV framework and support a more precise evaluation through simulation of various mixed traffic scenarios, thus facilitating AV integration in the real world in a faster and safer way.

## References


1. Aeberhard, M., Kühbeck, T., Seidl, B., Friedl, M., Thomas, J., & Scheickl, O. (2015). *Automated driving with ros at bmw*. Hamburg, Germany: ROSCon.
2. AIMSUN: Modelling connected vehicles using aimsun api: Cooperative adaptive cruise control (2017). https://www.aimsun.com/modelling-connected-vehicles-using-aimsun-api/
3. Amoozadeh, M., Deng, H., Chuah, C. N., Zhang, H. M., & Ghosal, D. (2015). Platoon management with cooperative adaptive cruise control enabled by VANET. *Vehicular communications, 2*(2), 110–123.
4. Arcile, J., Sobieraj, J., Klaudel, H., & Hutzler, G. (2017). Combination of simulation and model-checking for the analysis of autonomous vehicles' behaviors: A case study. In *Multi-agent systems and agreement technologies* (pp. 292–304). Berlin: Springer.
5. Barceló, J., & Casas, J. (2005). Dynamic network simulation with AIMSUN. In *Simulation approaches in transportation analysis* (pp. 57–98). Springer, Berlin.
6. Bashiri, M., Jafarzadeh, H., & Fleming, C.H. (2018). Paim: Platoon-based autonomous intersection management. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, (pp. 374–380). IEEE
7. Bast, H., Delling, D., Goldberg, A., Muller-Hannemann, M., Pajor, T., Sanders, P., Wagner, D., & Werneck, R.F. (2016). Route planning in transportation networks. In *Algorithm engineering*, (pp. 19–80). Springer, Berlin.
8. Boban, M., Manolakis, K., Ibrahim, M., Bazzi, S. & Xu, W. (2016). Design aspects for 5g v2x physical layer. In *IEEE Conference on Standards for Communications and Networking (CSCN)*, (pp. 1–7). IEEE, Berlin, Germany
9. Boesch, P.M. & Ciari, F. (2015). Agent-based simulation of autonomous cars. In *2015 American Control Conference (ACC)*, (pp. 2588–2592). IEEE
10. Caltagirone, L., Bellone, M., Svensson, L., & Wahde, M. (2019). Lidar camera fusion for road detection using fully convolutional neural networks. *Robotics and Autonomous Systems, 111,* 125–131.
11. Chen, S., Hu, J., Shi, Y., & Zhao, L. (2016). Lte-v: A td-lte-based v2x solution for future vehicular network. *IEEE Internet of Things journal, 3*(6), 997–1005.
12. Chen, Z., Zhang, J., & Tao, D. (2019). Progressive lidar adaptation for road detection. *IEEE/CAA Journal of Automatica Sinica, 6*(3), 693–702.
13. Contet, J.M., Gechter, F., Gruer, P., & Koukam, A. (2006). Multiagent system model for vehicle platooning with merge and split capabilities. In *Third International Conference on Autonomous Robots and Agents* (pp. 41–46)
14. Contet, J. M., Gechter, F., Gruer, P., & Koukam, A. (2011). Reactive multi-agent approach to local platoon control: stability analysis and experimentations. *International Journal of Intelligent Systems Technologies and Applications, 10*(3), 231–249.
15. Cvišić, I., Ćesić, J., Marković, I., & Petrović, I. (2018). Soft-slam: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles. *Journal of Field Robotics, 35*(4), 578–595.
16. Davila, A., & Nombela, M. (2012). Platooning —safe and eco-friendly mobility. Tech. Rep. 2012-01-0488, SAE Technical Paper
17. DeBoeser, C. (2019). Towards a Hierarchical State-Machine-Based Framework for Platoon Manoeuvre Descriptions. Master's thesis, Technical University of Munich, Munich, Germany
18. Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik, 1*(1), 269–271.
19. Dolgov, D., Thrun, S., Montemerlo, M., & Diebel, J. (2008). Practical search techniques in path planning for autonomous driving. In *Proceedings of the First International Symposium on Search Techniques in Artificial Intelligence and Robotics* (STAIR-08). AAAI, Chicago, USA
20. Dong, Y., Xu, Y., Li, D., Hu, C., & Xi, Y. (2019). Model of platoon evolution in lvm scenario. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, (pp. 2932–2937). IEEE

21. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., & Koltun, V. (2017). Carla: An open urban driving simulator. arXiv preprint arXiv:1711.03938
22. El Najjar, M. E., & Bonnifait, P. (2005). A road-matching method for precise vehicle localization using belief theory and kalman filtering. *Autonomous Robots, 19*(2), 173–191.
23. Emery, J., Boyard-Micheau, J., Marilleau, N., Martiny, N., & Thévenin, T. (2013). Exploitation of traffic counting data for traffic study in urban areas: from traffic assignment to simulation model validation. In *18th European Colloquium in Theoretical and Quantitative Geography (ECTQG)*. Dourdan, France. https://hal.archives-ouvertes.fr/hal-00881294
24. Ericson, C. (2004). *Real-time collision detection*. Boca Raton: CRC Press.
25. Fellendorf, M., & Vortisch, P. (2010). Microscopic traffic flow simulator vissim. In J. Barceló (Ed.), *Fundamentals of traffic simulation, international series in operations research & management science* (pp. 63–93). Berlin: Springer.
26. Fridman, L., Terwilliger, J., & Jenik, B. (2018). Deeptraffic: Crowdsourced hyperparameter tuning of deep reinforcement learning systems for multi-agent dense traffic navigation. arXiv preprint arXiv:1801.02805
27. Furda, A., & Vlacic, L. (2011). Enabling safe autonomous driving in real-world city traffic using multiple criteria decision making. *IEEE Intelligent Transportation Systems Magazine, 3*(1), 4–17.
28. Gechter, F., Contet, J. M., Galland, S., Lamotte, O., & Koukam, A. (2012). Virtual intelligent vehicle urban simulator: Application to vehicle platoon evaluation. *Simulation Modelling Practice and Theory, 24,* 103–114.
29. Gechter, F., Koukam, A., Debain, C., Dafflon, B., El-Zaher, M., Aufrère, R., et al. (2020). Platoon control algorithm evaluation: Metrics, configurations, perturbations, and scenarios. *Journal of Testing and Evaluation, 48*(2), 1478–1504.
30. Gipps, P. G. (1981). A nehavioural car-following model for computer simulation. *Transportation Research Part B: Methodological, 15*(2), 105–111.
31. Gong, S., Shen, J., & Du, L. (2016). Constrained optimization and distributed computation based car following control of a connected and autonomous vehicle platoon. *Transportation Research Part B: Methodological, 94,* 314–334.
32. González, D., Pérez, J., Milanés, V., & Nashashibi, F. (2016). A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems, 17*(4), 1135–1145.
33. GROUP, P.: Virtual testing of autonomous vehicles with ptv vissim (2018). https://www.ptvgroup.com/en/solutions/products/ptv-vissim/areas-of-application/autonomous-vehicles-and-new-mobility/
34. Guériau, M., Dafflon, B., & Gechter, F. (2017). Vips: A simulator for platoon system evaluation. *Simulation Modelling Practice and Theory, 77,* 157–176.
35. Hamdar, S. H., Treiber, M., Mahmassani, H. S., & Kesting, A. (2008). Modeling driver behavior as sequential risk-taking task. *Transportation research record, 2088*(1), 208–217.
36. Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics, 4*(2), 100–107.
37. Helbing, D., & Tilch, B. (1998). Generalized force model of traffic dynamics. *Physics Review E, 58,* 133–138. https://doi.org/10.1103/PhysRevE.58.133.
38. Helly, W. (1959) Simulation of bottlenecks in single-lane traffic flow
39. Henderson, T. R., Lacage, M., Riley, G. F., Dowell, C., & Kopena, J. (2008). Network simulations with the ns-3 simulator. *SIGCOMM demonstration, 14*(14), 527.
40. Hoogendoorn, R., van Arem, B., & Hoogendoorn, S. (2014). Automated driving, traffic flow efficiency, and human factors. *Journal of the Transportation Research Board, 2422*(1), 113–120.
41. Hörl, S., Becker, F., & Axhausen, K. W. (2020). Automated mobility on demand: A comprehensive simulation study of cost, behaviour and system impact for zurich. *Arbeitsberichte Verkehrs-und Raumplanung, 1534,*.
42. Horni, A., Nagel, K., & Axhausen, K. (Eds.). (2016). *Multi-agent transport simulation MATSim*. London: Ubiquity Press. https://doi.org/10.5334/baw.
43. Huang, Y., Ding, H., Zhang, Y., Wang, H., Cao, D., Xu, N., & Hu, C. (2019). A motion planning and tracking framework for autonomous vehicles based on artificial potential field elaborated resistance network approach. *IEEE Transactions on Industrial Electronics, 67*(2), 1376–1386.
44. Huang, Y.K., Chen, Z.W., & Lee, C. (2009). Continuous k-nearest neighbor query over moving objects in road networks. In *Advances in Data and Web Management*, pp. 27–38. Springer: Berlin.
45. Huang, Z., Chu, D., Wu, C., & He, Y. (2018). Path planning and cooperative control for automated vehicle platoon using hybrid automata. *IEEE Transactions on Intelligent Transportation Systems, 20*(3), 959–974.
46. IEEE: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Std 802.11-2016, IEEE (2016)

47. Ilas, C. (2013). Electronic sensing technologies for autonomous ground vehicles: A review. 8th International symposium on Advanced Topics in Electrical Engineering (ATEE) pp. 1–6
48. Issariyakul, T., & Hossain, E. (2009) Introduction to network simulator 2 (ns2). In*Introduction to network simulator NS2*, (pp. 1–18). Springer: NewYork.
49. Ivanchev, J., Braud, T., Eckhoff, D., Zehe, D., Knoll, A., & Sangiovanni-Vincentelli, A. (2019). On the Need for Novel Tools and Models for Mixed Traffic Analysis. In *26th International Transport System World Congress*. Singapore: NewYork.
50. Ivanchev, J., Knoll, A., Zehe, D., Nair, S., & Eckhoff, D. (2019). A Macroscopic Study on Dedicated Highway Lanes for Autonomous Vehicles. In *International Conference on Computational Science (ICCS)*. Springer, Faro, Portugal
51. Jia, D., Lu, K., Wang, J., Zhang, X., & Shen, X. (2015). A survey on platoon-based vehicular cyber-physical systems. *IEEE communications surveys & tutorials, 18*(1), 263–284.
52. Kalra, N., & Paddock, S. M. (2016). Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice, 94,* 182–193.
53. Karunasekera, H., Wang, H., & Zhang, H. (2019) Multiple object tracking with attention to appearance, structure, motion and size. IEEE Access
54. Kato, S., Tokunaga, S., Maruyama, Y., Maeda, S., Hirabayashi, M., Kitsukawa, Y., Monrroy, A., Ando, T., Fujii, Y., & Azumi, T. (2018). Autoware on board: Enabling autonomous vehicles with embedded systems. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pp. 287–296. IEEE
55. Kavathekar, P., & Chen, Y. (2011) Vehicle platooning: A brief survey and categorization. In *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, (pp. 829–845). American Society of Mechanical Engineers
56. Kesting, A., Treiber, M., & Helbing, D. (2007). General lane-changing model mobil for car-following models. *Transportation Research Record, 1999*(1), 86–94.
57. Kesting, A., Treiber, M., & Helbing, D. (2007). General lane-changing model MOBIL for car-following models. *Transportation Research Record: Journal of the Transportation Research Board, 1999*(1), 86–94.
58. Kesting, A., Treiber, M., & Helbing, D. (2010). Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 368*(1928), 4585–4605.
59. Ketenci, U., Brémond, R., Auberlet, J. M., Grislin, E., et al. (2014). Drivers with limited perception: model and application to traffic simulation. *Recherche Transports Sécurité, 30,* 47–61.
60. Khayatian, M., Mehrabian, M., Andert, E., Dedinsky, R., Choudhary, S., Lou, Y., & Shirvastava, A. (2020). A survey on intersection management of connected autonomous vehicles. *ACM Transactions on Cyber-Physical Systems, 4*(4), 1–27.
61. Khodayari, A., Ghaffari, A., Nouri, M., Salehinia, S., & Alimardani, F. (2012). Model predictive control system design for car-following behavior in real traffic flow. In *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*, (pp. 87–92). IEEE
62. Kohli, P., & Chadha, A. (2018). Enabling pedestrian safety using computer vision techniques: A case study of the 2018 uber inc. self-driving car crash. arXiv preprint arXiv:1805.11815
63. Krajzewicz, D., Hertkorn, G., Rössel, C., & Wagner, P. (2002). Sumo (simulation of urban mobility); an open-source traffic simulation. In *4th Middle East Symposium on Simulation and Modelling (MESM 2002)*, pp. 183–187. Sharjah, UAE
64. Kuwata, Y., Teo, J., Fiore, G., Karaman, S., Frazzoli, E., & How, J. P. (2009). Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology, 17*(5), 1105–1118.
65. Larson, J., Liang, K. Y., & Johansson, K. H. (2015). A distributed framework for coordinated heavy-duty vehicle platooning. *IEEE Transactions on Intelligent Transportation Systems, 16*(1), 419–429.
66. Leonard, J., How, J., Teller, S., Berger, M., Campbell, S., Fiore, G., et al. (2008). A perception-driven autonomous urban vehicle. *Journal of Field Robotics, 25*(10), 727–774.
67. Levinson, J., & Thrun, S. (2010). Robust vehicle localization in urban environments using probabilistic maps. In *IEEE International Conference on Robotics and Automation*, (pp. 4372–4378). Anchorage, Alaska
68. Liang, C. Y., & Peng, H. (1999). Optimal adaptive cruise control with guaranteed string stability. *Vehicle System Dynamics, 32*(4–5), 313–330.
69. Luo, L. .h, Liu, H., Li, P., & Wang, H. (2010). Model predictive control for adaptive cruise control with multi-objectives: comfort, fuel-economy, safety and car-following. *Journal of Zhejiang University SCIENCE A, 11*(3), 191–201.

70. Mahmassani, H. S. (2016). Autonomous vehicles and connected vehicle systems:flow and operations considerations. *Transportation Science, 50*(4), 1140–1162.
71. Maiti, S., Winter, S., & Kulik, L. (2017). A conceptualization of vehicle platoons and platoon operations. *Transportation Research Part C: Emerging Technologies, 80,* 1–19.
72. Marin-Plaza, P., Hussein, A., Martin, D., & Escalera, A.d.l. (2018). Global and local path planning study in a ros-based research platform for autonomous vehicles. Journal of Advanced Transportation **2018**
73. Mattas, K., Makridis, M., Hallac, P., Raposo, M. A., Thiel, C., Toledo, T., & Ciuffo, B. (2018). Simulating deployment of connectivity and automation on the antwerp ring road. *IET Intelligent Transport Systems, 12*(9), 1036–1044.
74. Michon, J.A. (1985). A critical view of driver behavior models: what do we know, what should we do? In *Human behavior and traffic safety*, (pp. 485–524). Springer: NewYork.
75. Milanés, V., Shladover, S. E., Spring, J., Nowakowski, C., Kawazoe, H., & Nakamura, M. (2013). Cooperative adaptive cruise control in real traffic situations. *IEEE Transactions on intelligent transportation systems, 15*(1), 296–305.
76. Ntousakis, I.A., Nikolos, I.K., & Papageorgiou, M. (2014). On microscopic modelling of adaptive cruise control systems. In ISTS14, vol. 6, pp. 111–127. Elsevier, Corsica, France
77. Pendleton, S., Andersen, H., Du, X., Shen, X., Meghjani, M., Eng, Y., et al. (2017). Perception, planning, control, and coordination for autonomous vehicles. *Machines, 5*(1), 6.
78. Petrillo, A., Salvi, A., Santini, S., & Valente, A. S. (2018). Adaptive multi-agents synchronization for collaborative driving of autonomous vehicles with multiple communication delays. *Transportation Research part C: Emerging Technologies, 86,* 372–392.
79. Ploeg, J., Scheepers, B.T., Van Nunen, E., Van de Wouw, N., & Nijmeijer, H. (2011). Design and experimental evaluation of cooperative adaptive cruise control. In *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 260–265. Qingdao, China
80. Rajamani, R. (2011). Vehicle dynamics and control. Springer Science & Business Media
81. Robinson, T., Chan, E., & Coelingh, E. (2010). Operating platoons on public motorways: An introduction to the sartre platooning programme. In *17th World Congress on Intelligent Transport Systems*, vol. 1, p. 12. Busan, South Korea
82. Rosique, F., Navarro Lorente, P., Fernandez, C., & Padilla, A. (2019). A systematic review of perception system and simulators for autonomous vehicles research. *Sensors, 19,* 648.
83. Schoenberg, J.R., Nathan, A., & Campbell, M. (2010). Segmentation of dense range information in complex urban scenes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2033–2038. Taipei, Taiwan
84. Segata, M., Joerer, S., Bloessl, B., Sommer, C., Dressler, F., & Cigno, R.L. (2014). Plexe: A platooning extension for veins. In *IEEE Vehicular Networking Conference (VNC)*, pp. 53–60. Paderborn, Germany
85. Sheikholeslam, S., & Desoer, C.A. (1990). Longitudinal control of a platoon of vehicles. In *1990 American Control Conference*, (pp. 291–296). IEEE, San Diego, CA, USA
86. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., & Li, H. (2019) Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. arXiv preprint arXiv:1912.13192
87. Shladover, S., Lu, X.Y., Yang, S., Ramezani, H., Spring, J., Nowakowski, C., & Nelson, D. (2018). Cooperative adaptive cruise control (cacc) for partially automated truck platooning
88. Shladover, S. E. (2007). PATH at 20 - History and major milestones. *IEEE Transactions on intelligent transportation systems, 8*(4), 584–592.
89. Shladover, S. E., Su, D., & Lu, X. Y. (2012). Impacts of cooperative adaptive cruise control on freeway traffic flow. *Transportation Research Record, 2324*(1), 63–70.
90. Siraj, S., Gupta, A., & Badgujar, R. (2012). Network simulation tools survey. *International Journal of Advanced Research in Computer and Communication Engineering, 1*(4), 199–206.
91. Sommer, C., Eckhoff, D., Brummer, A., Buse, D.S., Hagenauer, F., Joerer, S., & Segata, M. (2019). Veins: The open source vehicular network simulation framework. In *Recent Advances in Network Simulation*,( pp. 215–252)
92. Sommer, C., German, R., & Dressler, F. (2011). Bidirectionally coupled network and road traffic simulation for improved ivc analysis. *IEEE Transactions on Mobile Computing, 10*(1), 3–15.
93. de Souza, F., Verbas, O., & Auld, J. (2019). Mesoscopic traffic flow model for agent-based simulation. *Procedia Computer Science, 151,* 858–863.
94. Taillandier, P., Gaudou, B., Grignard, A., Huynh, Q. N., Marilleau, N., Caillou, P., et al. (2019). Building, composing and experimenting complex spatial models with the gama platform. *GeoInformatica, 23*(2), 299–322.

95. Talebpour, A., & Mahmassani, H. S. (2016). Influence of connected and autonomous vehicles on traffic flow stability and throughput. *Transportation Research Part C: Emerging Technologies, 71,* 143–163.
96. Talebpour, A., Mahmassani, H. S., & Hamdar, S. H. (2015). Modeling lane-changing behavior in a connected environment: A game theory approach. *Transportation Research Procedia, 7,* 420–440.
97. Tian, B., Deng, X., Xu, Z., Zhang, Y., & Zhao, X. (2019). Modeling and numerical analysis on communication delay boundary for cacc string stability. *IEEE Access, 7,* 168870–168884.
98. Treiber, M., Hennecke, A., & Helbing, D. (2000). Congested Traffic States in Empirical Observations and Microscopic Simulations. *Physical Review E, 62*(2), 1805–1824.
99. Treiber, M., Kesting, A., & Helbing, D. (2006). Delays, inaccuracies and anticipation in microscopic traffic models. *Physica A: Statistical Mechanics and its Applications, 360*(1), 71–88.
100. Tsugawa, S., Jeschke, S., & Shladover, S. E. (2016). A review of truck platooning projects for energy savings. *IEEE Transactions on Intelligent Vehicles, 1*(1), 68–77.
101. Van Arem, B., Van Driel, C. J., & Visser, R. (2006). The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Transactions on Intelligent Transportation Systems, 7*(4), 429–436.
102. Wang, D., Pan, Q., Zhao, C., Hu, J., Liu, L., & Tian, L. (2016). Slam-based cooperative calibration for optical sensors array with gps/imu aided. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, (pp. 615–623). IEEE
103. Wang, H., Wang, R., Wu, T., Zhu, K., & Zhang, L. (2019). ipcs: intersection platoon control scheme for non-traffic light community. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, (pp. 3502–3507). IEEE
104. Wegener, A., Piórkowski, M., Raya, M., Hellbrück, H., Fischer, S., & Hubaux, J.P. (2008). Traci: an interface for coupling road traffic and network simulators. In *11th Communications and Networking Simulation Symposium (CNS)*, (pp. 155–163). ACM
105. Weyns, D., Steegmans, E., & Holvoet, T. (2004). Towards active perception in situated multi-agent systems. *Applied Artificial Intelligence, 18*(9–10), 867–883. https://doi.org/10.1080/08839510490509063.
106. Wolcott, R.W., & Eustice, R.M. (2014). Visual localization within lidar maps for automated urban driving. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (pp. 176–183). Chicago, IL, USA
107. Wongpiromsarn, T., Topcu, U., & Murray, R.M. (2010). Receding horizon control for temporal logic specifications. In *13th ACM International Conference on Hybrid Systems: Computation and Control (HSCC)*, (pp. 101–110). Stockholm, Sweden
108. Zehe, D., Nair, S., Knoll, A., & Eckhoff, D. (2017). Towards CityMoS: A Coupled City-Scale Mobility Simulation Framework. In *5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication (FG-IVC 2017)*, (pp. 26–28). FAU Erlangen-Nuremberg, Erlangen, Germany
109. Zhang, J., & Singh, S. (2014). Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems Conference(RSS)*. Berkeley, CA
110. Zhang, J., & Singh, S. (2015). Visual-lidar odometry and mapping: Low drift, robust, and fast. In *IEEE International Conference on Robotics and Automation(ICRA)*. Seattle, WA
111. Ziegler, J. (2014). Making Bertha Drive - An Autonomous Journey on a Historic Route. *IEEE Intelligent Transportation Systems Magazine, 6*(2), 8–20.